

# Handwritten Sentence Recognition

U.-V. Marti and H. Bunke

Institut für Informatik und angewandte Mathematik  
Universität Bern, Neubrückstrasse 10, CH-3012 Bern  
Switzerland

email: {marti,bunke}@iam.unibe.ch

## Abstract

*In this paper we present a system for reading handwritten sentences and paragraphs. The system's main components are preprocessing, feature extraction and recognition. In contrast to other systems, whole lines of text are the basic units for the recognizer. Thus the difficult problem of segmenting a line of text into individual words can be avoided. Another novel feature of the system is the incorporation of a statistical language model into the recognizer. Experiments on the database described in [8] have shown that a recognition rate on the word level of 79.5% and 60.05% for small (776 words) and larger (7719 words) vocabularies can be reached. These figures increase to 84.3% and 67.32% if the top ten choices are taken into regard.*

## 1 Introduction

In the last years the field of handwriting recognition was the topic of intensive research. While the first systems read segmented characters, later systems aimed at the recognition of cursively handwritten words. Only short time ago the first systems appeared which are able to read sequences of words. Typical applications of word sequence recognition are address [5] or check reading [6, 2], where recognizers operate in a small, specific domain. At the moment only very few systems are known which address the domain of free text recognition [7]. Typically, these systems segment the text into words. But as it is known from the field of continuous speech recognition [9], the segmentation of complete sentences into single words is difficult and prone to errors. In particular, it is hard to recover in the recognition phase from errors that occurred during the earlier stage of segmentation.

In the present paper we propose a system that treats complete handwritten lines of text as basic input units. Segmentation of a line of text into individual words is obtained as a byproduct of recognition, which is based on hidden Markov

models (HMMs). Another novel feature of the proposed system is the use of a statistical language model, which is incorporated in the recognizer to improve its performance. The system was trained and tested on the database described in [8]. This database consists of a collection of about 5030 lines of text written by about 250 writers. The total number of word instances in the database is 44019 words.

Section 2 gives a description of the system. This section is divided into four subsections describing, preprocessing, feature extraction, recognition, and the statistical sentence model. In Section 3 experimental results are reported. In particular, it is shown how the system behaves under different sizes of the vocabulary. Finally, in Section 4, we draw conclusions from this work.

## 2 System Description

The system described in this paper consist of three main components: preprocessing, feature extraction, and recognition using HMMs.

In addition to these three main components, a bigram language model [3] is used. It not only drives the recognition of the individual words in a line of text, but it also improves the recognition results. The database used to derive the language model is the Lancaster-Oslo/Bergen corpus [4], a collection of texts from different domains. This corpus contains about 500 English texts classified into 15 text categories, i.e. popular lore, belles letters, general fiction and so on. Each text contains about 2000 words. Totally there are about 1'000'000 words in the whole corpus.

### 2.1 Preprocessing

The original data input to the system are images of complete pages of handwritten text from the underlying database. An example is shown in Fig. 1. The database includes a few tools to extract the individual lines from a page [8]. An example is shown in Fig. 2. Given a line of text, different normalization operations are carried out. The main steps

Today, for example, the Foreign Minister of Indonesia arrived in Belgrade as the guest of the Yugoslav Foreign Minister. In fact such Yugoslav activity has been ~~pro~~ particularly intensified in the past year or so and though so far, apart from joint action in the United Nations, these exchanges have not been seen on any wider basis, President Tito is known for some time to have favoured a conference of neutralist leaders.

**Figure 1. Image of a text page.**

been seen on any wider

**Figure 2. Image of an uncorrected line fragment of text (Fig. 1, line 3 from bottom).**

consists of skew and slant correction, positioning and scaling of the text lines.

In the first step the skew of the considered text line is corrected. For each image column the lowest black pixel is determined. Thus the lower contour of the writing is obtained. The skew angle of the line can then be measured by regression analysis on this set of points [10]. Once the skew angle is determined the line can be rotated into horizontal position.

After deskewing, a slant correction is done. Here we measure the angle between the writing and the vertical direction. For this purpose, the contour of the writing is approximated by small lines. The directions of these lines are accumulated in an angle histogram. The angle corresponding to the maximum value in the histogram gives the slant [1]. After the slant angle has been determined, a shear operation brings the writing in an upright position (see Fig. 3).

For the vertical positioning of the text line, the lower baseline determined during skew correction serves as line of reference. Given this line, a scaling procedure is applied. For this procedure, we need to additionally know the upper baseline. It is computed by horizontal projection of the text line. To the histogram of black pixels resulting from the horizontal projection, an ideal histogram is fitted. From this ideal histogram the position of the upper baseline is ob-

been seen on any wider

**Figure 3. Image of a corrected line fragment.**

tained. The bounding box of a line of text together with the upper and lower baseline define three disjoint areas (upper, middle, and lower). Each of these areas is scaled in vertical direction to a predefined size of equal height.

For horizontal scaling the black-white transitions in the considered line of text are counted. This number of transitions can be set in relation to the mean transition number, which is determined over the whole image database off-line. Thus the scaling factor for the horizontal direction is obtained.

All preprocessing operations described above, in particular positioning and scaling, are required to make the feature extraction procedure described in the next section properly working.

## 2.2 Feature Extraction

To extract a sequence of features from a text line, a sliding window is used. A window of one column width and the image's height is moved from left to right over each text line. (Thus there is no overlap between two consecutive window positions.) To determine the features at each window position nine geometrical characteristics are computed.

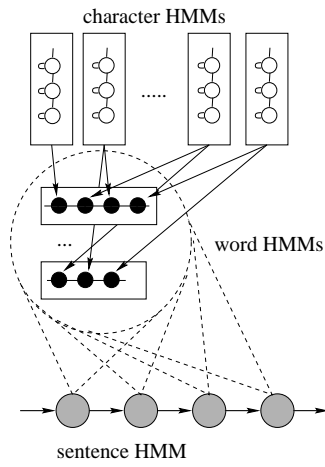
The first three features are the weight of the window (i.e. the number of black pixels), its center of gravity and the second order momentum of the window. This set characterizes the window from a more global point of view. It describes how many pixels in which region of the window are, and how they are distributed.

Features four to nine give more details about the writing. Features four and five define the position of the upper and the lower contour in the window. The next two features, number six and seven, give the orientation of the upper and the lower contour in the window by the gradient of the contour at the window's position. As feature number eight the number of black-white transitions in vertical direction is used. Finally, feature number nine, gives the number of black pixels between the upper and lower contour.

Notice that all these features can be easily computed from the binary image of a text line. However to make the features robust against different writing styles, careful preprocessing as described in Section 2.1 is necessary.

## 2.3 Hidden Markov Models

Hidden Markov Models (HMMs) are widely used in the field of pattern recognition. Their original application was



**Figure 4. Recognition network.**

in speech recognition [9]. But because of the similarities between speech and handwriting recognition, HMMs have become very popular in handwriting recognition as well.

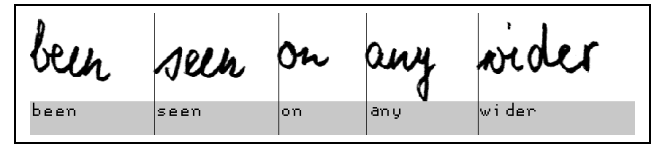
In systems with a small vocabulary, it is possible to build an HMM for each word. But for large vocabularies this method doesn't work anymore because not enough training data is available. Therefore, in our system an HMM is build for each character. The use of character models allows to share training data. Each instance of a letter in the training set has an impact on the training and leads to a better parameter estimation.

To achieve optimal recognition results, the character HMMs have to be fitted to the problem. In particular the number of states, the possible transitions and the output probability distributions has to be choosen.

In our system each character model consists of 14 states. This number has been found empirically. Because of the left to right direction of writing, a linear transition structure has been choosen for the character models. From each state only the same or the succeeding state can be reached. Because of the continuous nature of the features, the output probability distributions  $b(o|s_i)$  are continous. To adjust the transition probabilities  $p(\cdot)$  and the output probability distributions  $b(\cdot)$  during training the Baum-Welch algorithm [9] is used.

In the recognition phase the character models are concatenated to words, and the words to sentences. Thus a recognition network is obtained (see Fig. 4). In this network the best path can be found with the Viterbi algorithm [9]. It corresponds to the desired recognition result. I.e., the best path represents the sequence of words with maximum probability, given the image of the input sentence.

One crucial feature of the system described in this paper is that text lines are not segmented into single words during preprocessing, but the segmentation of the line into words



**Figure 5. The recognition and segmentation result of a text line fragment.**

is delivered by the HMM as a byproduct of the recognition process (see Fig. 5).

## 2.4 Statistical Language Model

In natural language, the frequency of words is not equally distributed. Neither are position and sequence of words random. Therefore, additional knowledge about the language can be introduced in the recognition of handwritten sentences.

In the system described in this paper, a unigram and a bi-gram language model has been introduced. These models weight each word by its occurrence probability. To compute these probabilities, we used the LOB-corpus, which contains about 1'000'000 word instances [4]. In particular, the following three quantities are obtained from the corpus:  $N$  - the total number of word instances of the considered vocabulary found in the corpus;  $N_i$  - the number of word instances of word  $i$  of the vocabulary;  $N_{i,j}$  - the number of instances of word pair  $(i, j)$  occurring in the corpus.

From these numbers the following probabilities can be computed:

$$p(i) = \frac{N_i}{N} \quad (1)$$

is the probability that word  $i$  occurs, and

$$p(j|i) = \frac{N_{i,j}}{N_i} \quad (2)$$

is the probability that word  $j$  follows word  $i$ .

During Viterbi decoding, these probabilities weight the words in the vocabulary. The unigram probability  $p(i)$  is applied to a word at the beginning of a sentence where no contextual knowledge is available. Later bigram probabilities  $p(j|i)$  are applied. They induce knowledge about preceding words into the recognition task.

## 3 Experiments

In our experiments the database described in [8] was used. This database includes image data of handwritten text of several hundred writers, which have written whole sentences and paragraphs; see Fig 1. The images in the

| System        | Vocabulary | Word inst. | Rec.Rate | 10-Best |
|---------------|------------|------------|----------|---------|
| <i>u</i>      | 776 words  | 2212       | 71.34%   | 76.69%  |
| <i>s</i>      | 412 words  | 4523       | 79.50%   | 84.30%  |
| <i>a</i>      | 1487 words | 4815       | 54.69%   | 60.70%  |
| <i>ab</i>     | 2703 words | 11000      | 49.33%   | 55.08%  |
| <i>abc</i>    | 3411 words | 14806      | 46.79%   | 52.76%  |
| <i>abcdef</i> | 4409 words | 21462      | 61.68%   | 67.64%  |
| <i>a-r</i>    | 7719 words | 44019      | 60.05%   | 67.32%  |

**Table 1. Recognition results under different vocabulary size**

database can be grouped into sets with different characteristics depending, for example, on the number of words in the vocabulary, or on the number of different writers. In our experiments we consider the number of words in the vocabulary as main characteristic. A summary of the experimental results is given in Tab. 1. The number of different writers varies from 1 in system *u* to approximately 300 in system *a-r*. The data set of each experiment with smaller vocabulary is a subset of the larger sets, except systems *u* and *s*. In all experiments the training sets contain 4/5 of the chosen data. The rest is used to test the system.

In the first experiment (system *u*), a small set, which only contains the writing of one single writer, is used. There are 572 lines of text and 2212 word instances out of a vocabulary of 776 words. A word recognition rate of 71.34% is achieved in the top choice, and 76.69% if the top ten choices are considered.

Also the second set (system *s*) is rather small, but the texts were written by six different persons. There are 541 lines, including 4523 word instances. The vocabulary contains 412 words. A word recognition rate of 79.5% in the top and 84.3% in the ten best choices is reached. Because more training data is available, the parameter estimation of the HMM is more general, what leads to better recognition results.

The next experiments have much larger vocabularies, from 1487 words up to 7719 words. Also the number of lines which are used to train and test the different systems increases from 564 lines in System *a* up to 2461 lines in System *abcdef* and 5030 lines in System *a-r*. The number of writers in these sets is not exactly known; they can only be estimated. Assuming that one person has filled about two forms, a total of about 250 different writers result. The word recognition rate in these five systems lies between 46.79% (system *abc*) and 61.68% (system *abcdef*) depending on the vocabulary and the size of the training set (see Tab. 1). If the top ten choices are regarded, between 52.76% (system *abc*) and 67.64% (system *abcdef*) of the words are recognized correctly.

## 4 Conclusion

In this paper a system for recognizing handwritten sentences is presented. Methods used before in continuous speech recognition and single word handwriting recognition have been applied to free handwritten sentence recognition. In comparison to other systems, which segment the text into single words, this system treats complete lines of text as basic units. Moreover, because language has not a random character, linguistic knowledge is introduced by means of unigram and bigram models in order to improve the recognition performance. In future versions of the system, more sophisticated language models and larger vocabularies will be considered.

## References

- [1] T. Caesar, J. M. Gloger, and E. Mandler. Preprocessing and feature extraction for a handwriting recognition system. In *Proc. of the 2nd Int. Conf. on Document Analysis and Recognition, Tsukuba Science City, Japan*, pages 408–411, 1993.
- [2] N. Gorski, V. Anisimov, E. Augustin, D. Price, and J.-C. Simon. A2iA Check Reader: A Family of Bank Check Recognition Systems. In *5th Int. Conference on Document Analysis and Recognition 99, Bangalore, India*, pages 523–526, 1999.
- [3] F. Jelinek. Self-organized language modeling for speech recognition. In A. Waibel and K.-F. Lee, editors, *Readings in Speech Recognition*, pages 450–506. Morgan Kaufmann Publishers, Inc., 1990.
- [4] S. Johansson, G. Leech, and H. Goodluck. *Manual of Information to accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with digital Computers*. Department of English, University of Oslo, Oslo, 1978.
- [5] A. Kaltenmeier, T. Caesar, J. Gloger, and E. Mandler. Sophisticated topology of hidden markov models for cursive script recognition. In *Proc. of the Second Int. Conf. on Document Analysis and Recognition*, pages 139–142, 1993.
- [6] G. Kaufmann and H. Bunke. A system for the automated reading of check amounts - some key ideas. In *Proc. 3rd IAPR Workshop on Document Analysis Systems, Nagano, Japan*, pages 302 – 315, 1998.
- [7] G. Kim, V. Govindaraju, and S. Srihari. Architecture for Handwritten Text Recognition Systems. In S.-W. Lee, editor, *Advances in Handwriting Recognition*, pages 163–172. World Scientific, 1999.
- [8] U.-V. Marti and H. Bunke. A full English sentence database for off-line handwriting recognition. In *5th Int. Conference on Document Analysis and Recognition 99, Bangalore, India*, pages 705–708, 1999.
- [9] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.
- [10] M. Schüssler and H. Niemann. A hmm-based system for recognition of handwritten address words. In *Proceedings of Sixth Int. Workshop on Frontiers in Handwriting Recognition 98, Taejon, South Korea*, pages 505–514, 1998.